MATH 338: Homework #7

Due: Tuesday, April 9, 2019

Solve the below "project" related to simulating Markov chains.

This problem concerns simulating a Wright-Fisher model for the neutral evolution of an allele in the population. It will require you to simulate a stochastic process via software of your choice (some code will be provided below in MATLAB). Please read Section 4.3 of the notes for more information.

1. Write down the transition matrix A for the Wright-Fisher model with $2N = 4$. Note that this should be a 5×5 matrix.

2. Simulate **one** realization of the Markov chain for 10 generations, assuming that the population starts with 3 A alleles. Below is the almost complete code; the only aspect that is missing is the definition of the transition matrix A, which should come from part (a), and the initial condition.

```
%Clear all previous plots and variables
clear all; close all;

% Define the one-step transition matrix
% e.g. matrix definition in MATLAB
% A = [1 1; 0 1]; will define the 2x2 with ones in 1st row, and 0 1 in 2nd row
A = ;

% How many time units to simulate
tf = 10;

% Vector to store one realization
X = zeros(tf, 1);

% Initial number of A alleles (initial condition)
x0  = ;
% Note that in the matrix, this corresponds to the 4th row
X(1) = x0 + 1;

% Simulate Markov chain via transition probabilities
for i=2:tf
    current_state = X(i-1);
    U = rand; % uniform random number between 0 and 1
    transition_row = A(current_state,:);  % This is the row you are transitioning from
    bound = 0;
    % Idea:  Generate a uniform random number, and choose a state based on
    % that number.  The weights come from the entries (of the appropriate
    % row) of the transition matrix.
    for new_state = 1:length(transition_row)
        if U <= bound + transition_row(new_state)
            X(i) = new_state;
```

```
            break;
        else
            bound = bound + transition_row(new_state);
        end
    end
end

% X currently takes values 1 to 5.  Normalize to go from 0 to 4.
X = X-1;

% Plot the results
figure
plot(X, '-xb', 'LineWidth',2)
xlabel('time');
ylabel('X');
axis([1 tf 0 4]);
title('Realization of Wright-Fisher with 2N=4');
```

Type this into a MATLAB editor, save it, and then you can just run it from the software directly (there is a "Run" button). Note: you **should NOT** type this into the command window directly, as this is not good practice. Of course, you can use any software you'd like, if you are more comfortable translating the above to a different language.

Provide the plot for **one** realization of the Wright-Fisher chain.

3. Simulate **10000** realizations of the chain, and **numerically** compute the expected value of $X(10)$,

$$\mathbb{E}(X(10)).$$

Note: you should be able to adapt the above code in a minor way to store all simulations. Basically, you need to wrap the entire simulation procedure in another "for" loop, which runs from 1 to 10000. X should then be a matrix, with 10000 columns:

```
    X = zeros(tf,10000);
```

and for the appropriate index, you update X as

```
    X(i, i_sim_index) = new_state;
```

Here i_sim_index is the index keeping track of the current realization. Note that you can compute expected values using the "mean" MATLAB function:

```
expected_value = mean(X(tf,:));  % This takes the average at the final time across all
                                 %simulations
```

4. What is the **theoretical** expected value of $X(10)$? How does this compare with your result from Problem 3? Note that you should use MATLAB to compute the necessary matrix powers.

5. Similarly to Problem 3, **numerically** compute the variance of $X(10)$. Note that, similarly to "mean," there is a MATLAB command to do this.

6. Using conditioning, derive an expression for the **theoretical** variance of $X(10)$. Compare this with your result from Problem 5.